International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

Mapping Complex, Large – Scale Spiking Networks on Neural VLSI

Christian Mayr[1], Matthias Ehrlich[1], Stephan Henker[1], Karsten Wendt, and René Schüffny

***Abstract*—**Traditionally, VLSI implementations of spiking neural nets have featured large neuron counts for fixed computations or small exploratory, configurable nets. This paper presents the system architecture of a large configurable neural net system employing a dedicated mapping algorithm for projecting the targeted biology-analog nets and dynamics onto the hardware with its attendant constraints.

***Keywords*—**Large scale VLSI neural net, topology mapping, complex pulse communication.

I. Introduction

VLSI implementations of pulse coupled neural nets aimed at exploring various computational aspects of biological neural nets have so far mainly explored two avenues:

On the one hand, networks have been created with simple dynamics and fixed, repetitive network structures with very little flexibility, but relatively high neural element count [1], [2]. The processing function(s) of these nets have been determined a priori, emulating cut-outs from biological structures and operations, either to analyze/understand these functions or use them in technical application. The limited flexibility of these designs relegates them to large-scale proof-of-concept of a certain functionality, while further exploration of said functionality necessitates an IC redesign.

The other avenue of exploration consists of IC's employing complex topologies and neuron/synapse dynamics with attendant large configuration memories. These have been relegated to small nets (100-1000 neurons, <10k Synapses), even if the hardware has been designed such as to permit the linking of several chips [3],[4]. These nets are primarily used for exploring network and element behavior, and as a supplement/complement to software-based neuro-simulators. The scientific target of confirming or analyzing data or behavioral models delivered by the neuro-theoretic or neurobiology community is the main objective in this case. Because of the full individual reconfigurability of neuroelements on the IC (i.e. synapses, neurons) and the low element count (permitting full connectivity among all elements via flexible electronic axons and dendrites), the transfer of net topologies and element configuration is trivial from an organizational/structural point of view. The only challenge for these ICs would be to project biology-centric neuro-variables such as membrane potential, conduction delays, leakage terms, adaptation constants, etc to their electronic representations on the IC.

The work presented in [5] is a step towards a kind of system achieving a synthesis of both approaches, with large element count and flexible, yet hardware-constrained configuration. However, the element count could still be improved, and more complex synapse and neuron dynamics realized.

In this paper, we present a system architecture currently under development that will allow very large (>1e6 neurons, >1e9 synapses) reconfigurable networks to be built, in the form of interlinked Dies on a single wafer. Hardware constraints are identified and a description of the mapping software is given which is needed to faithfully reproduce biological network structures with their attendant plasticity in VLSI. The efficacy of the mapping algorithm is documented via a few samples of the topology mapping.

The complete system will be used as a research tool for exploring various computational paradigms as postulated from neurobiological evidence.

II. System Description

If we want to keep the flexibility and configurable network dynamics of complex nets needed to explore new computational properties on VLSI chips, but also extend the processing to very large nets, some compromise has to be achieved between reconfigurability and VLSI hardware constraints.

Hardware design is generally constrained by the available resources, especially chip area. Considering a small hardware implementing 10^3 neurons with 10^3 synapses each requires a crossbar with $10^3 x 10^6$ switches and configuration memory of 10MBit to allow full flexibility. For implementation, already this small example is not feasible and the proposed hardware is implementing orders of magnitude more neural elements.

Hardware constraints in general encompass the following:

1) not enough configuration memory for neuro-elements, so this memory has to be shared, with synapses/neurons

Manuscript received December 21st, 2006. The authors thank the European Union for the financial support in the framework of the Information Society Technologies program, project FACETS (Nr. 15879).

All authors are with the Circuits and Systems Laboratory, Department of Electrical Engineering, Dresden University of Technology, Dresden, Germany (corresponding author is Matthias Ehrlich, phone: +49-351-463-36930; fax: +49-351-463-37260; e-mail: ehrlich@iee.et.tu-dresden.de).

(1) The first three authors contributed equally to the research described herein.

having similar parameters grouped around this shared memory.

2) not enough configuration memory and IC space for electronic axons and dendrites, so a full network connectivity cannot be achieved.
3) the dynamics of neurons and synapses have to be achieved using less IC space, which inherently speeds up operation of these elements (e.g. smaller integration capacitors), so any analysis circuits and the supply/biasing backbone of such an IC would have to be that much faster
4) the increased speed of these elements also leads to increased communication bandwidth (both intra-chip and off chip for analysis and linking of IC's)

The FACETS hardware platform is proposed for a 1M neurons implemented on several interconnected wavers. The wavers are not diced but used completely by connecting the individual reticles on the wafer by wafer scale interconnect. The reticles consist of configurable analogue network core ASIC´s, which finally encompass the neurons, synapses and connection structures. Each neuron in this system connects in average to 1k other neurons via plastic synaptic links. The research undertaken at TU Dresden is to design the systems communication backplane, the hardware systems simulation and the configuration of the system together with the necessary benchmarks.

A. Communication Architecture

The Analogue Network Core - ANC of which a schematic can be seen in Fig. 1 forms the basic element of the FACETS Architecture.

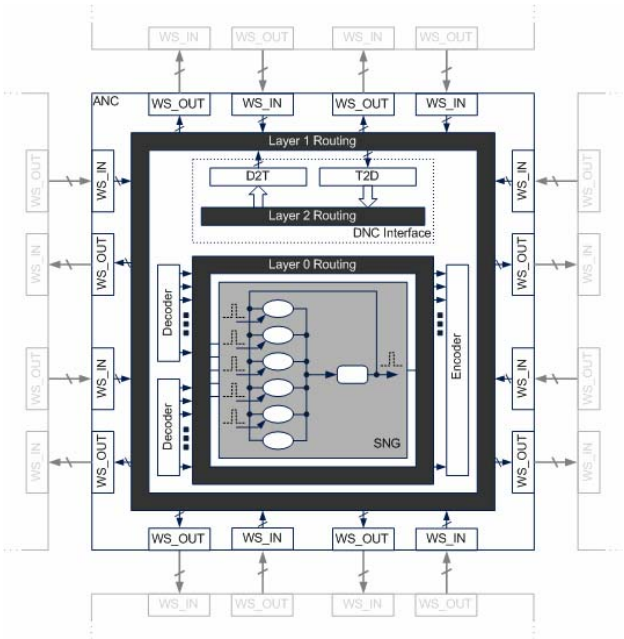

Fig. 1 ANC Schematic

The logical architecture of the FACETS system communication is a resource constrained three layered structure.

The smallest configuration units of the system are Synapse-Neuron Groups directly linked together with a so called layer 0 connections. Around the analog core elements, the communication layer 1 of multiplexed continuous-time connections is implemented as ring bus structure.

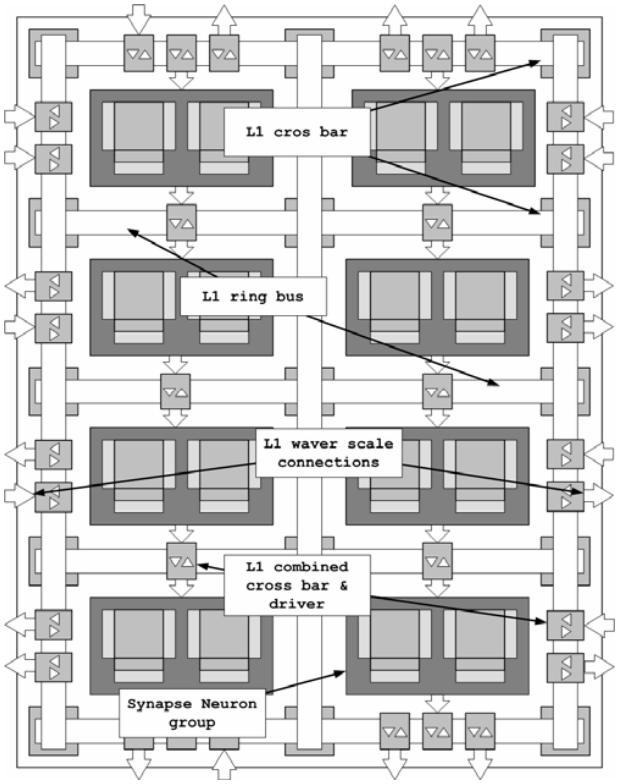

Fig. 2 Structural view of the L0/L1 communication

Layer 1 connections can be directed over die boundaries to adjacent ANCs and to the interface of communication layer 2. Layer 2 communication, configuration and general backplane is provided by a PCB backplane situated above the wafer. This backplane contains dedicated ASICs designed for interfacing with the ANCs, so called Digital Network Chips (DNCs). Fig. 3 will provide an overview of the concept.

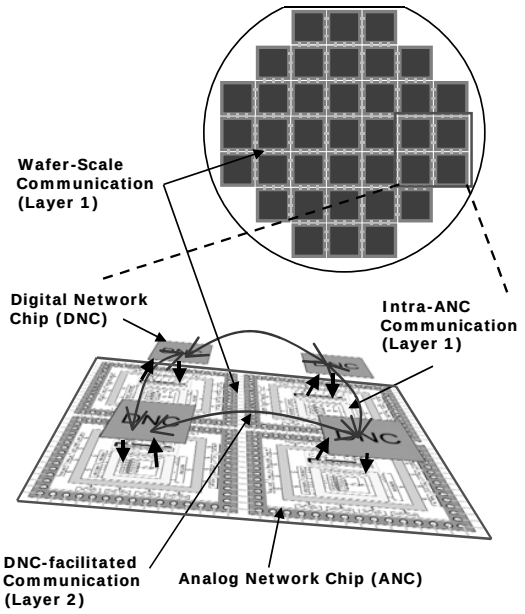

Fig. 3 Waferscale system overview

Layer 2 connections are used for long range connection to distant ANCs using a time discretized events-within-a-packet based protocol. The implementation of Layer 2 is utilizing high speed serial LVDS links also designed by the TU Dresden research group.

The three different layers can be distinguished by their communication paradigms and different constraints. The number of routings via a specific layer will be constraint by a limited number of connections per layer and the load dependency of those connections.

TABLE I
COMMUNICATION LAYER DESCRIPTION

Layer	Type	Description
0	continuous-time, analog direct connection	Dendritic compartments and their respective synapse groups are linked via configurable analog current connections to form compartmental neurons.
1	continuous-time, multiplexed	Main backbone of pulse communication, multiplexed asynchronous digital pulse communication. Load independent due to static routed connections, with a slight chance of pulse loss for colliding events. Connections to adjacent ANC's within a certain range of die 'hops' can be routed over this layer.
2	discrete, packed based	The number of connections is limited by the load capacity of the channels. In this case the activity of the routed connections determines the number of routable channels. Used also for external communication with/analysis of network.

B. Neural Elements

The synapses will be similar to [5] i.e. they will realize an STDP learning rule which can be modified via digital look-up-tables (LUT) to realize additive, multiplicative and power-law weight updates [6]. The parametrizable form of the weight update also makes it possible to let the synapses behave in a BCM-like fashion. Fast synaptic adaptations also form part of the plasticity available on the synapses [7]. Supervisor input or steered learning can be achieved via externally governed weight changes or forcing the neuron to fire at selected points in time [8].

The Hodgkin-Huxley-derived conductance-based IF-neurons [5] are implemented as sections of a dendrite, which can be connected in series from 4 to 64 dendritic compartments, with the spike traveling along the compartments via an analog bus.

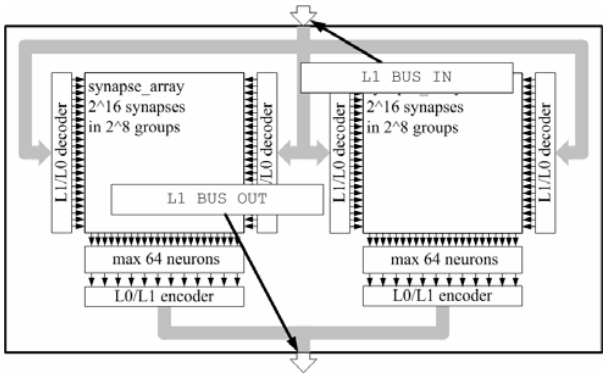

Fig. 4 One single ANC cell

As seen in Fig. 4, a sub-block of the ANC is composed of an array of 256 synapse groups, each consisting of 256 individual synapses. In turn, four synapse group are connected to a dendrite section, so the minimum number of synapses per neuron is 1024. The maximum number of synapses per neuron is 16384 if 16 dendritic compartments are connected in series. This range of neuronal fan-in is comparable to average cortical neurons [9]. The synapse groups share the LUT and the parameter storage, so only synapses with similar targeted behavior should be mapped on a single group. The pre- and postsynaptic time measurement and the synapse weight, however, are independent, so their dynamic time course is distinct for each synapse. Every synapse is configured for a single presynaptic neuron, whose output pulses are transmitted via the communication architecture as described above.

C. Mapping

The major problem that arises from the hardware-constraint-driven flexibility reduction is the mapping of experimental neural networks on the hardware resources. The task is to place and route given neurons and synapses to a configurable neuronal STDP array and to optimize the synaptic connections. Different connection routings influence the faithfulness of biology-reproduction, as well as the overall number of routable connections. In other words the routing should be done with a minimum of channels, as close to target parameters of a neural net (i.e. synapse and neuron adaptation

behaviour, delays, etc) as possible and with a minimum of routing costs.

III. Mapping Software

A. Mapping & Optimization Procedure

The mapping and optimization procedure is done in three steps with rising granularity. Note, steps do not correspond to the different layers. The configuration of the system itself can be defined as mapping task, whereas the optimization is a multi objective search for least cost mapping. The different optimization objectives are as follows:

5) To minimize the routing costs
6) To come as close as possible to the given network parameters. For this, maximum deviations from these target parameters have to be established
7) To use as little ANC´s as possible, that is to concentrate as much neurons as possible together.
8) To route connections with higher load over load independent Layers.
9) To route connections with a higher probability first (for probabilistic network descriptions

Basically, the algorithm maps the logical neurons of any given net to the physical neurons on the system, connects them according to the network topology and creates a matrix which reflects the connections over the complete system as show in the simplified example in Fig. 5. Different target functions apply for the granularity steps, as outlined below.

1) Step 1 – Hyper Global

For the uppermost mapping step, centering on the wafer-level system, the target is to concentrate as much synapses as possible on single wafers (i.e. squares on the main diagonal in the connection matrix), inter-wafer communication has to be reduced. All synapses realized between neurons on different wafers have the same penalty, reflecting the packet-based, all-to-all communication architecture between the wafers.

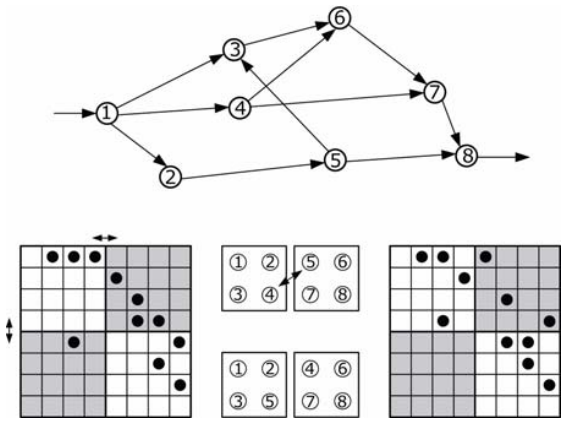

Fig. 5 Simple example of mapping a network to a connection matrix representation and optimization by reordering

As seen in the above example, a reordering in the assignment of neurons to a wafer can result in reduced connection density between the wafers.

The number of s-permutations, or variations #VAR which is the number of possible configurations without recurrences for a given number of neurons #MAX_NRN in a neural network to the number of physically available neurons #PHY_NRN can be calculated with:

$$\#VAR = \frac{(\#\text{MAX\_NRN})!}{(\#\text{MAX\_NRN}-\#\text{PHY\_NRN})!} \quad (1)$$

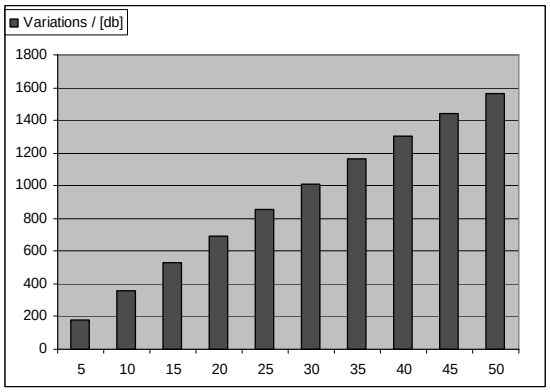

Fig. 6 Variations in [dB] for a 64 NRN array

So the Complexity of the problem makes it obviously impossible to probe all the possible ordering variations of the connection matrix as can be seen in Fig. 5 as number of variations without recurrences on a 64 NRN array with increasing net size, showing a linearized exponential growth.

Different heuristic algorithm and Sparse Matrix reordering algorithms like Reverse Cuthill McKee [1] where tested which yield up to 10% improvement in relative connection density for global routing on a network generated with uniform distribution of synaptic connections with 5 to 15 % connection density. On structured nets with a more regular structure, however, the same effort results in significantly higher optimization (Fig. 8).

2) Step 2– Global

The matrix can then be split up into sub blocks representing the single wavers to proceed to the next mapping step. In this step, the neurons are assigned to individual ANC's on the wafer, which are interconnected via post-processing waferconnects executed between single dies on the wafer. Here, the ordering of the ANCs also becomes an issue, i.e. the communication paths between dies cannot be treated as equal, as was the case for inter-wafer paths. The next figure gives an example of this:

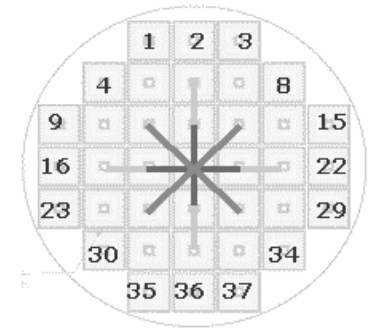

Fig. 7 37 ANCs on a single die, with nearest-neighbour, and longitudinal and diagonal single-hop Layer1 communication marked

If we take the system of Layer1 busses connecting the ANCs on the wafer, it is evident that a direct connect between two adjacent ANCs uses less bus resources than a extended connect which crosses one ANC. So this has to be taken into account for the mapping target in the global step. As a example for the global mapping step, Fig. 8 presents a simplified, layered, feed forward V1 according to [9] of 2500 neurons with 300k synapses.

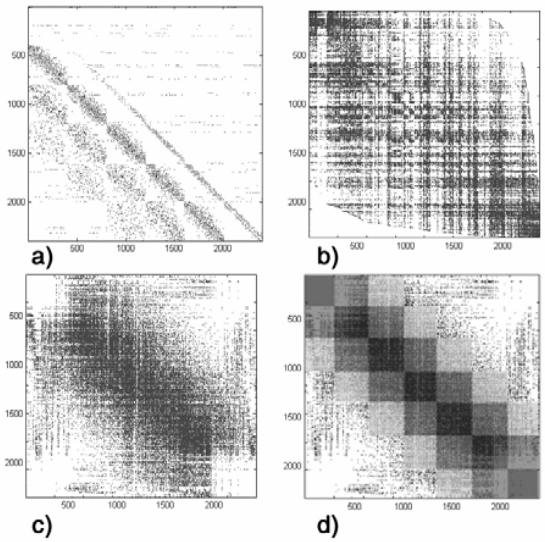

Fig. 8 Mapping example on part of a single wafer for a V1-analog network, initial V1 net (a), sorted with Reverse Cuthill McKee algorithm (b), GA reordering applied (c), and superimposed connection structure (d)

The above figure starts out with the initial V1 net as derived from [6], where a Reverse Cuthill McKee [10] reordering is shown to be insufficient for the task (b) due to the large number of outliers, caused by the feedforward-nature of the network (lower triangular structure) and the feedback pyramidal cells. A Travelling-Salesman GA is applied to the problem in (c), managing to center a substantial part of the feedforward connections on the center diagonal. Finally, in (d) the connection structure of the wafer is superimposed, with the different gray levels corresponding to the Layer1 connections between the ANCs as denoted in Fig. 7. Projecting the synapse connections on the different Layer1 connections gives feedback for the hardware development (i.e. where do bottlenecks develop?) and results in an approximate, high-level configuration of the hardware system for a given net. This approximate configuration is then passed on to the next step to reach a fine-granular, detailed configuration for each single ANC.

The next benchmark uses a less structured net generated from the low granularity, stochastic V1 model description given in [11]. On the hardware side, this benchmark is based on a generic waver-scale architecture, with 2 wavers, 4 ANCs/waver and 125 neurons/ANC, realizing a biological network with 1000 neurons and 135651 synapses. The effectiveness of the mapping algorithm can be seen from the following table, which gives percentages of total synapses realized via the different communication layers before and after application of the mapping.

TABLE II
MAPPING PERFORMANCE AS PER CENTAGE OF TOTAL SYNAPSES

	Layer 1	Waferscale	Layer 2
Initial Network	12.5	37.5	50.0
After Mapping	21.2	48.2	30.6

In general, the mapping algorithm achieves the most improvement for highly structured nets, such as cortical structures. It can still improve evenly connected networks somewhat, but is of course limited by the underlying entropy of the network structure. The above table gives results for an interim benchmark, which uses a layered V1 structure, but does not take into account any distance information between neurons, such as a decrease in connection probability with increase in the distance between neurons. Best mapping results can be obtained for a network with detailed structure such as the one in Fig. 8, which employs macro- as well micro-scale cortical geometry information.

3) Step 3– Local

The local optimization concentrates on single ANC´s. After finishing the basic partitioning in Step 2, the resulting connection matrix can again be split up into sub blocks representing single ANC's with the central part of connections inside the ANC itself and the vertical and horizontal part of outgoing and incoming connection of the selected the ANC. The available routing resources consist of Layer 1 and Layer 0 connections. Prior to optimisation, a first iteration is done by selecting connections with minimum costs until available resources are exhausted. The following iterations use the matrix swap algorithm as described above to reduce the local costs, and after convergence returns to global mapping with either successful mapping or with connections that could not be mapped. Global optimisation is then relocating those connections for a next local optimisation run.

Successful local mapping also generates the hardware configuration data which can be loaded onto the final

hardware to create the hardware representation of the original network.

IV. Data Handling

To give an overview of the amount of data and the complexity of the problem to be solved a short insight into implementation task shall be given. A rough estimation of the amounts of data to be handled shall lead to the problems to be solved in future work.

Taken as example a simple matrix, containing only the information on existing synaptic connections and utilizing a single bit to represent a connection a complete matrix will have ~1.2GByte. Alternatively a List representation storing only the nonzero elements (given above with a 100k Neurons x 1k synapses/per neuron) and the Elements x-y-matrix-index accordingly (which will need 17 bits per element assuming a 100k matrix) the list will need ~0.4GByte.

Although the latter gives a memory effort reduction by 2/3, the data handling becomes more difficult. A comparison of the effort accessing one element of the matrix vs. its list representation, utilizing an a priori classification of the costs and types of commands and their costs according to clock cycles and so on lead to the result that the cost of a list access are ~20 times the costs of the matrix access.

Due to memory limitations we decided at the moment for the List representation, accepting the longer processing time.

V. Conclusion/ Summary

We have presented a design effort targeted at implementing configurable large-scale neural networks in VLSI. This hardware will be used for confirming and extending simulation efforts on V1 and other mammalian cortex areas. Because of the faster network execution time for similar network sizes compared to software simulations (where state-of-the-art for $8*10^6$ neurons and $5*10^9$ synapses is execution in biological real time), developmental plasticity processes can be studied in detail.

A mapping software has been described which closes the gap between the hardware-constrained waferscale-system and biology-derived neural networks, matching up constraints such as (biological) axonal delays and (hardware) pulse routing delays, ensuring faithful reproduction of network behaviour.

During the next project steps we will gain precise information on the technology constraints and limits through research carried out in parallel on the hardware side. A first communication prototype was designed and is momentarily under fabrication.

A further step is to parallelize the optimization algorithms using POSIX Threads. Research is to be carried out on a two Opteron Dual Core Multi Processor system running on Linux. combining a locally threads and globally MPI model.

More effort will be expanded on the optimization algorithms, especially the application of multi-objective genetic algorithms for Hyper Global search which can be parallelized following the island model in [12].

The mapping will be extended to concentrate on further optimization parameters besides the communication, e.g. delays, pulse loss probability, etc.

References

[1] T. Morie, M. Nagata, and A. Iwata, “Design of a Pixel-Parallel Feature Extraction VLSI System for Biologically-Inspired Object Recognition Methods,” *Proc. International Symposium on Nonlinear Theory and its Application NOLTA ‘01*, pp. 371-374, 2001.

[2] J. Schreiter, U. Ramacher, A. Heittmann, D. Matolin, and R. Schüffny, “Cellular pulse coupled neural network with adaptive weights for image segmentation and its VLSI implementation,” in Proc. IS&T/SPIE 16th International Symposium on Electronic Imaging: Science and Technology, San Jose (CA), USA, 2004, 5298, pp. 290-296.

[3] G. Indiveri, E. Chicca, and R. Douglas, “A VLSI reconfigurable network of integrate–and–fire neurons with spike–based learning synapses,” *Proceedings of European Symposium on Artificial Neural Networks ESANN'2004*, pp. 405-410, 2004.

[4] S. Saighi, J. Tomas, Y. Bornat, and S. Renaud, “A conductance-based silicon neuron with dynamically tunable model parameters,” *Proceedings of Second International IEEE EMBS Conference on Neural Engineering*, pp. 285–288, 2005.

[5] J. Schemmel, K. Meier, and E Mueller, “A New VLSI Model of Neural Microcircuits Including Spike Time Dependent Plasticity,” *Proceedings of the 2004 International Joint Conference on Neural Networks IJCNN'04*, pp. 1711-1716, 2004.

[6] E. M. Izhikevich and N. S. Desai, "Relating STDP to BCM," *Neural Computation*, vol. 15, no. 7, pp. 151-1-1523, July 2003.

[7] H. Markram, Y. Wang, and M. Tsodyks, "Differential signaling via the same axon of neocortical pyramidal neurons," *Proc. Natl. Acad. Sci*., vol. 95, no. 9, pp. 5323-5328, April 1998.

[8] R. Legenstein, C. Näger, and W. Maass, "What can a neuron learn with spike-timing-dependent plasticity?," *Neural Computation*, vol. 17, no. 11, pp. 2337-2382, Nov. 2005.

[9] T. Binzegger, R. J. Douglas, and K. A. C. Martin, “A Quantitative Map of the Circuit of Cat Primary Visual Cortex,” *The Journal of Neuroscience*, pp. 8441-8453, 2004.

[10] Y.Saad, *Iterative Methods for Sparse Linear Systems*. Boston MA: PWS Publishing 1996.

[11] S. Haeusler and W. Maass, “A Statistical Analysis of Information-Processing Properties of Lamina-Specific Cortical Microcircuit Models,” in *Cerebral Cortex*, vol. 17, no. 1, pp. 149–162, 2007.

[12] Y. Chen, Z. Nakao, and X. Fang, “A Parallel Genetic Algorithm Based on the Island Model for Image Restoration,” *Proceedings of the 1996 IEEE Signal Processing Society Workshop* , pp. 109-118, 1996.